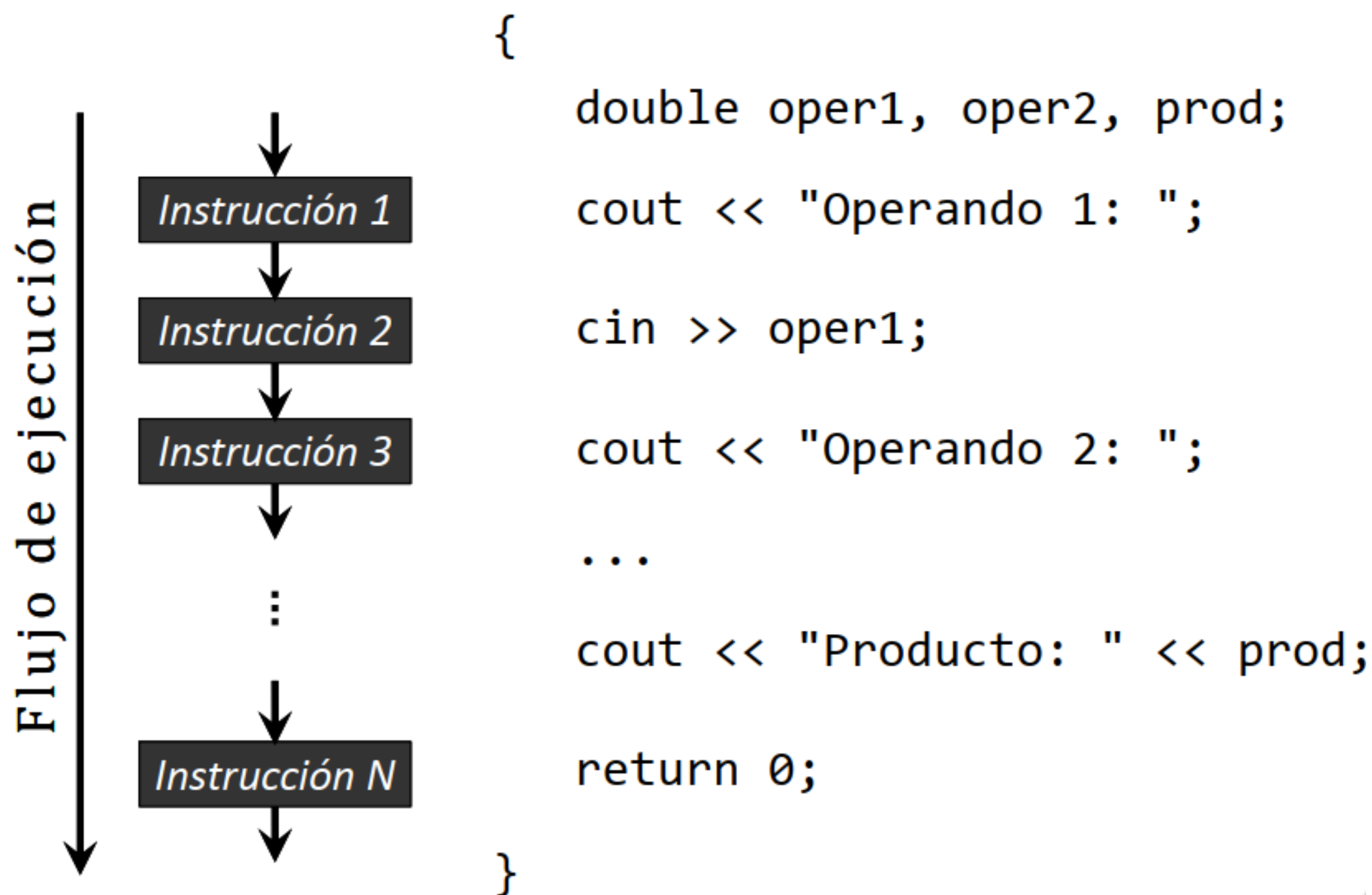


Toma de decisiones Condicionales



Flujo de ejecución

Ejecución secuencial



Selección simple

Selección simple (bifurcación)

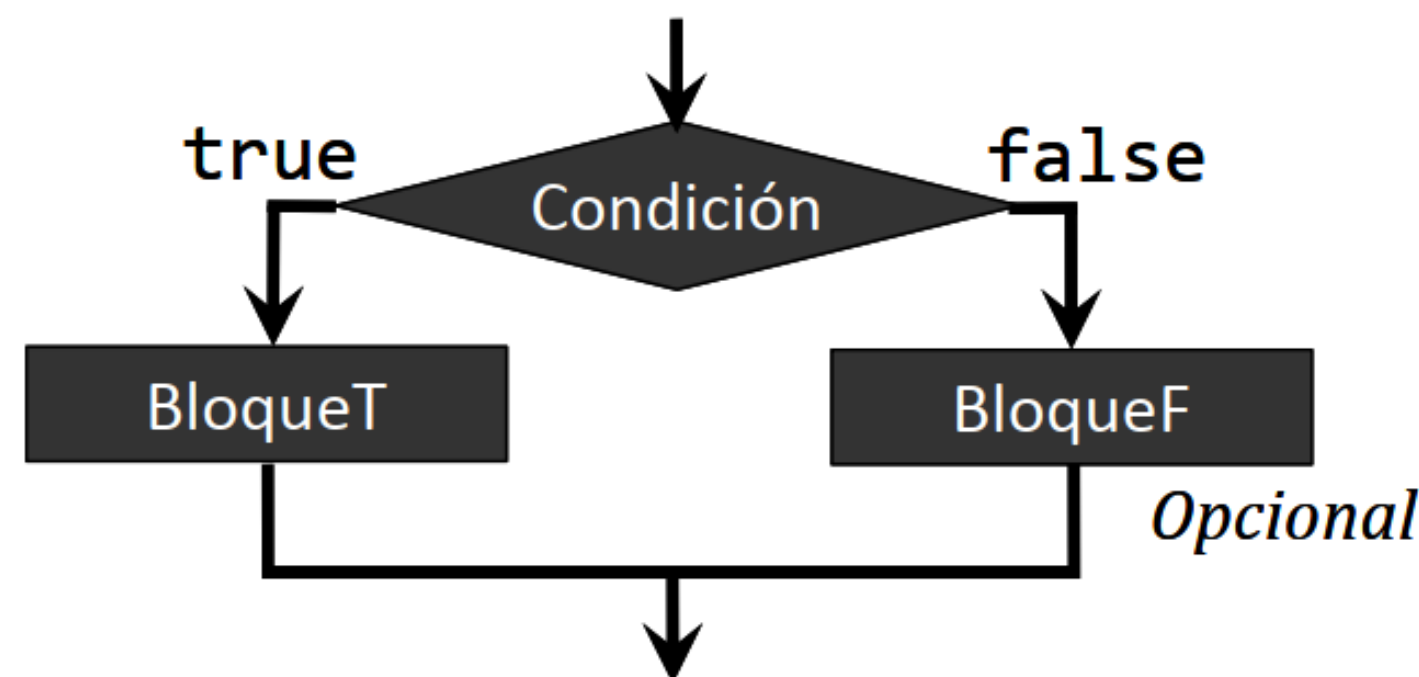


La instrucción if

```
if (condición) {
    ↪ códigoT
}
[else {
    ↪ códigoF
}]
```

condición: expresión bool

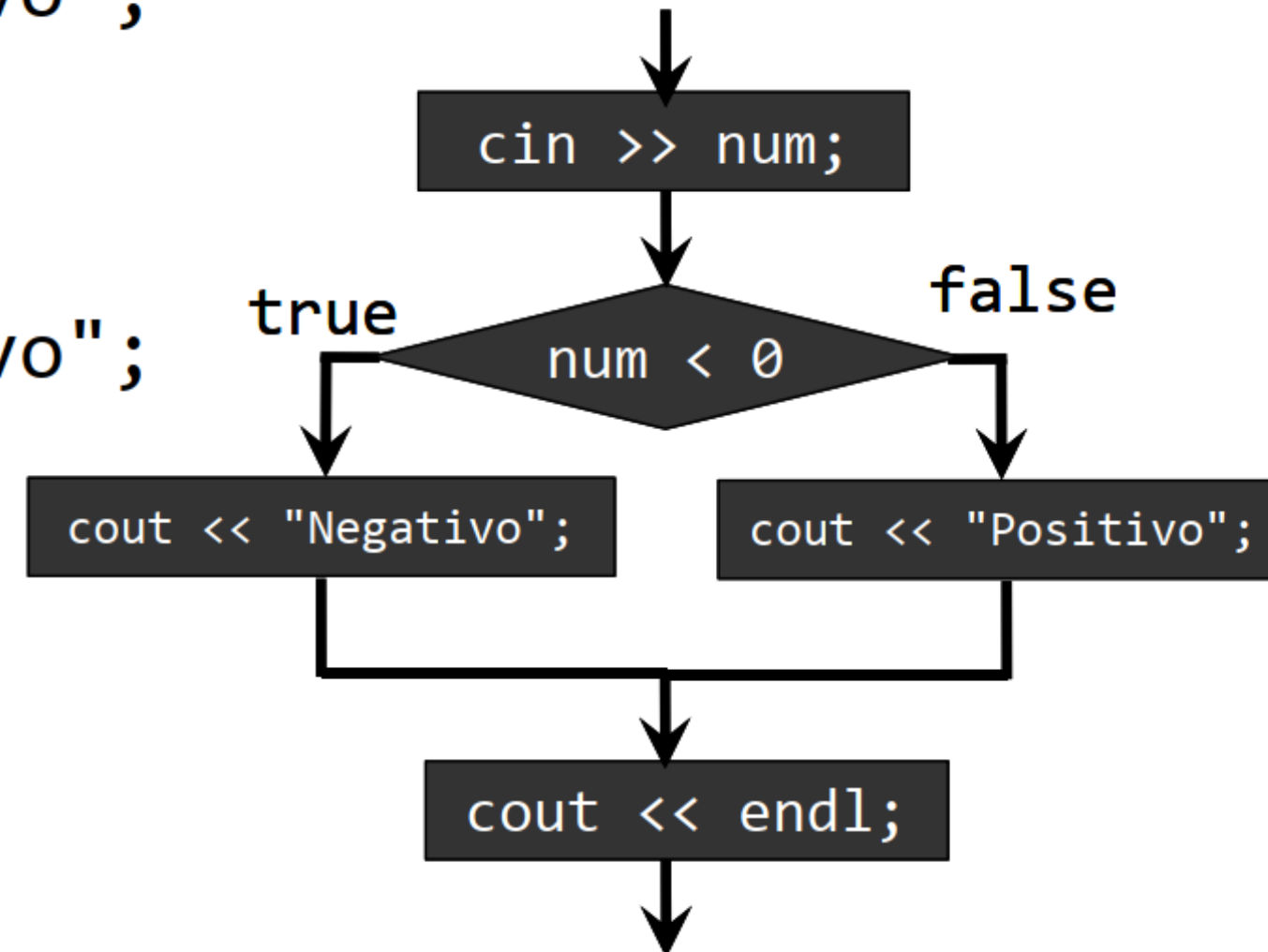
Cláusula else opcional



La instrucción if

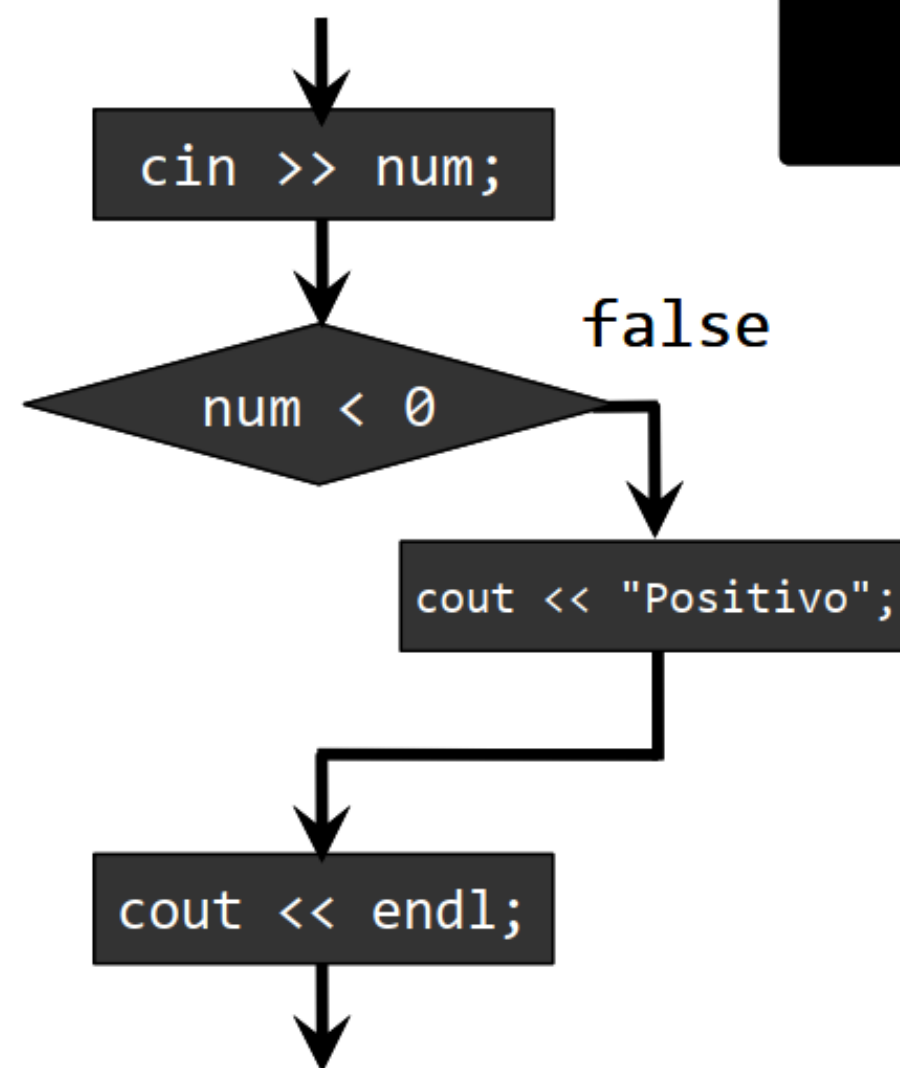
signo.cpp

```
int num;
cin >> num;
if (num < 0) {
    cout << "Negativo";
}
else {
    cout << "Positivo";
}
cout << endl;
```



La instrucción if

```
int num;
cin >> num;
if (num < 0) {
    cout << "Negativo";
}
else {
    cout << "Positivo";
}
cout << endl;
```

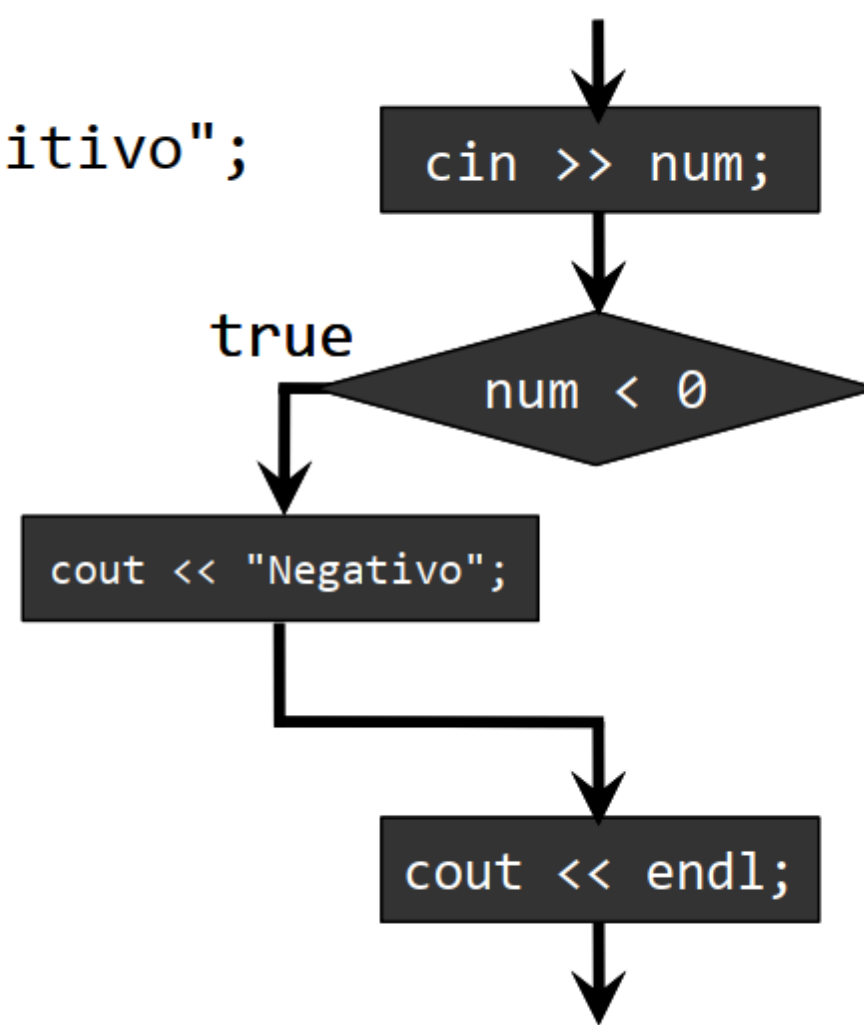


129
Positivo

num 129

La instrucción if

```
int num;
cin >> num;
if (num < 0) {
    cout << "Negativo";
}
else {
    cout << "Positivo";
}
cout << endl;
```

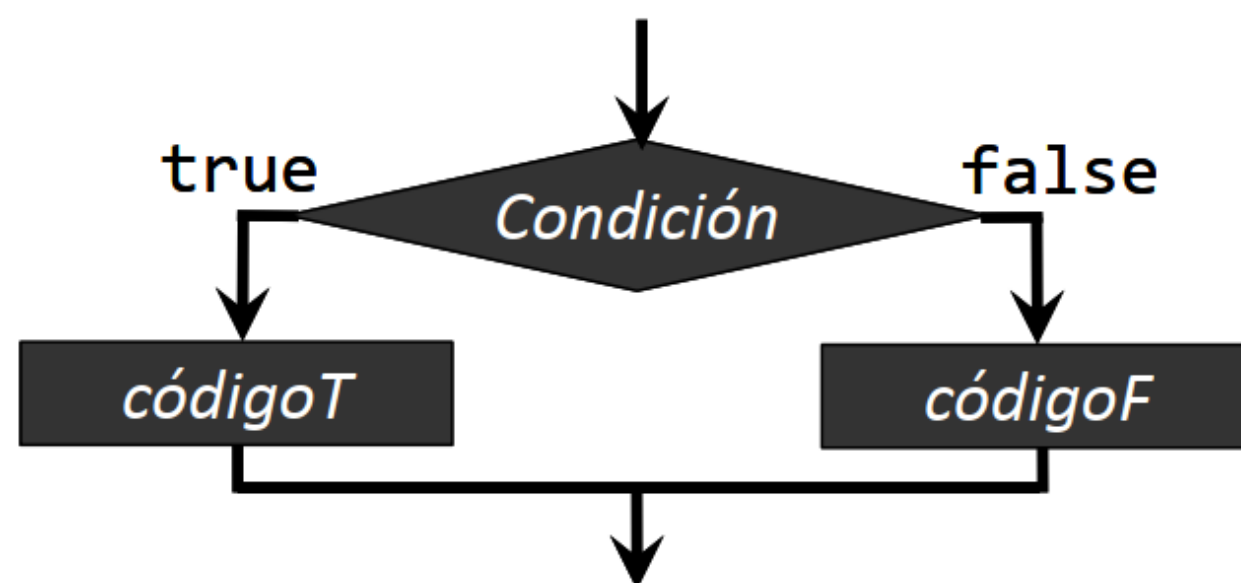


```
-5
Negativo
—
```

num -5

Hacer esto... o hacer esto otro...

Selección: bifurcación condicional



```

int num;
cout << "Número: ";
cin >> num;
if (num % 2 == 0) {
    cout << num << " es par";
}
else {
    cout << num << " es impar";
}
  
```

```

if (condición) {
    ↪ códigoT
}
  
```

```

else {
    ↪ códigoF
}
  
```

Opcional: puede no haber else

La instrucción `if`

selección.cpp

```
#include <iostream>
using namespace std;

int main() {
    int op1 = 13, op2 = 4;
    int opcion;
    cout << "1 - Sumar" << endl;
    cout << "2 - Restar" << endl;
    cout << "Opción: ";
    cin >> opcion;
    if (opcion == 1) {
        cout << op1 + op2 << endl;
    }
    else {
        cout << op1 - op2 << endl;
    }
    return 0;
}
```

```
D:\FP\Tema02>selección
1 - Sumar
2 - Restar
Opción: 1
17

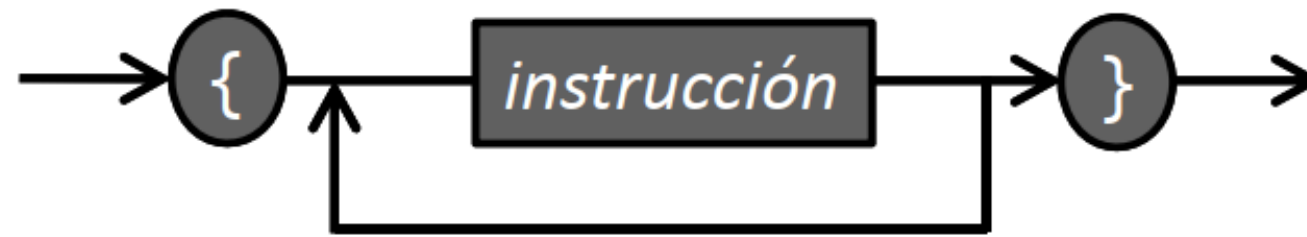
D:\FP\Tema02>selección
1 - Sumar
2 - Restar
Opción: 2
9
```

Bloques de código

Bloques de código

Agrupación de instrucciones

Grupo de instrucciones a ejecutar en una rama del if



```
int num, total = 0;
cin >> num;
if (num > 0)
```

```
{
    cout << "Positivo";
    total = total + num;
}
```

```
cout << endl;
```

```
{
  Tab ó 3 esp. → | instrucción1
                  | instrucción2
                  | ...
                  | instrucciónN
}
```

Ámbito local
(declaraciones locales)

Bloques de código

Posición de las llaves: cuestión de estilo

```
if (num > 0)
{
    cout << "Positivo";
    total = total + num;
}
cout << endl;
```

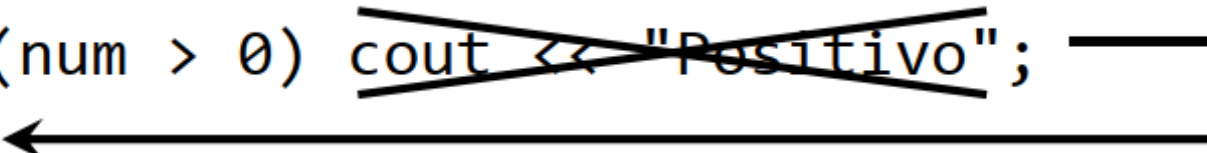
```
if (num > 0) {
    cout << "Positivo";
    total = total + num;
}
cout << endl;
```

No necesitamos las llaves si sólo hay una instrucción

```
if (num > 0) {
    cout << "Positivo";
}
≡
if (num > 0)
    cout << "Positivo";
```

Usaremos siempre llaves por simplicidad...

Evita poner el `if` y la instrucción objetivo en la misma línea:

```
if (num > 0) cout << "Positivo"; 
```

Ejemplo

división.cpp

División entre dos números protegida frente a intento de división por 0

```
#include <iostream>
using namespace std;

int main() {
    double numerador, denominador, resultado;
    cout << "Numerador: ";
    cin >> numerador;
    cout << "Denominador: ";
    cin >> denominador;
    if (denominador == 0) {
        cout << "Imposible dividir entre 0!";
    }
    else {
        resultado = numerador / denominador;
        cout << "Resultado: " << resultado << endl;
    }
    return 0;
}
```

Operadores lógicos

(condiciones compuestas)

< <= > >=

== !=

&&

||

Condiciones

`true || X ≡ true`

`(n == 0) || (x >= 1.0 / n)`

Si n es 0: ¿división por cero? (segunda condición)

Como la primera sería true: ¡no se evalúa la segunda!

`false && X ≡ false`

`(n != 0) && (x < 1.0 / n)`

Si n es 0: ¿división por cero? (segunda condición)

Como la primera sería false: ¡no se evalúa la segunda!

Ejemplo

condiciones.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int num;
    cout << "Introduce un número entre 1 y 10: ";
    cin >> num;
    if ((num >= 1) && (num <= 10)) {
        cout << "Número dentro del intervalo de valores válidos";
    }
    else {
        cout << "Número no válido!";
    }
    return 0;
}
```

¡Encierra las condiciones
simples entre paréntesis!

Condiciones equivalentes

```
((num >= 1) && (num <= 10))
((num > 0) && (num < 11))
((num >= 1) && (num < 11))
((num > 0) && (num <= 10))
```

Anidamiento de `if`

Número de días de un mes

diasmes.cpp

```
int mes, anio, dias;
cout << "Número de mes: ";
cin >> mes;
cout << "Año: ";
cin >> anio;
if (mes == 2) {
    if (bisiesto(mes, anio)) {
        dias = 29;
    }
    else {
        dias = 28;
    }
}
else {
    if ((mes == 1) || (mes == 3) || (mes == 5) || (mes == 7)
        || (mes == 8) || (mes == 10) || (mes == 12)) {
        dias = 31;
    }
    else {
        dias = 30;
    }
}
```

Asociación de cláusulas else

Cada `else` se asocia al `if` anterior más cercano sin asociar (mismo bloque)

```

if (condición1) {
    if (condición2) {...}
    else {...}
}
else {
    if (condición3) {
        if (condición4) {...}
        if (condición5) {...}
        else {...}
    }
    else { ...

```

Una mala sangría puede confundir

```

if (x > 0) {
    if (y > 0) {...}
else {...}

```



```

if (x > 0) {
    if (y > 0) {...}
else {...}

```



La sangría ayuda a asociar los `else` con sus `if`

La escala `if-else-if`

La escala if-else-if

Ejemplo:

Calificación (en letras)
de un estudiante en base
a su nota numérica (0-10)



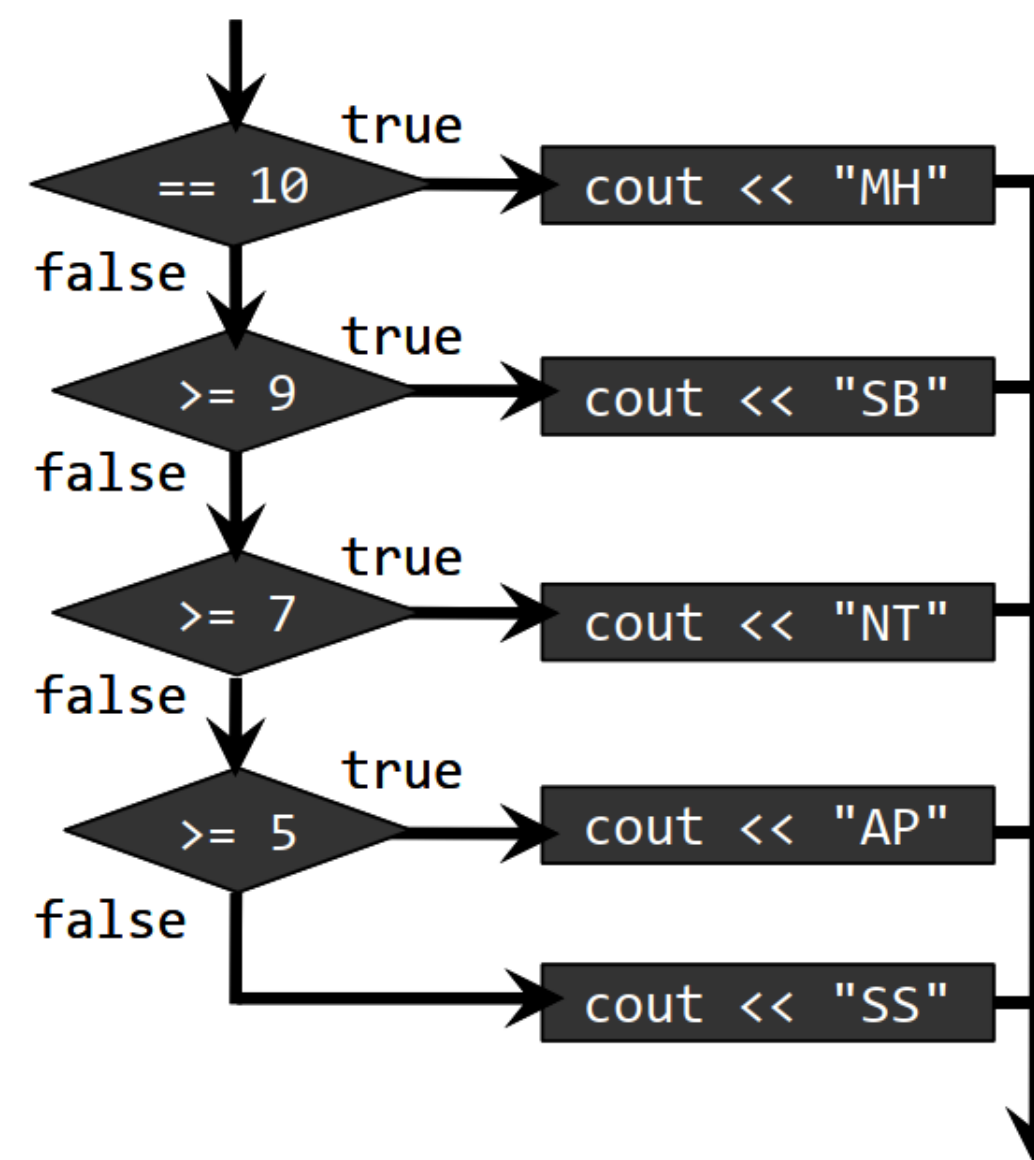
Si nota == 10 entonces MH

si no, si nota >= 9 entonces SB

si no, si nota >= 7 entonces NT

si no, si nota >= 5 entonces AP

si no SS



La escala if-else-if

nota.cpp

```
double nota;
cin >> nota;
if (nota == 10) {
    cout << "MH";
}
else {
    if (nota >= 9) {
        cout << "SB";
    }
    else {
        if (nota >= 7) {
            cout << "NT";
        }
        else {
            if (nota >= 5) {
                cout << "AP";
            }
            else {
                cout << "SS";
            }
        }
    }
}
```

≡

```
double nota;
cin >> nota;
if (nota == 10) {
    cout << "MH";
}
else if (nota >= 9) {
    cout << "SB";
}
else if (nota >= 7) {
    cout << "NT";
}
else if (nota >= 5) {
    cout << "AP";
}
else {
    cout << "SS";
}
```


La escala if-else-if

¡Cuidado con el orden de las condiciones!

```
double nota;
cin >> nota;
if (nota < 5) { cout << "SS"; }
else if (nota < 7) { cout << "AP"; }
else if (nota < 9) { cout << "NT"; }
else if (nota < 10) { cout << "SB"; }
else { cout << "MH"; }
```



```
double nota;
cin >> nota;
if (nota >= 5) { cout << "AP"; }
else if (nota >= 7) { cout << "NT"; }
else if (nota >= 9) { cout << "SB"; }
else if (nota == 10) { cout << "MH"; }
else { cout << "SS"; }
```

¡No se ejecutan nunca!



Sólo muestra AP o SS

Nivel de un valor

nivel.cpp

```
#include <iostream>
using namespace std;
int main() {
    int num;
    cout << "Introduce el nivel: ";
    cin >> num;
    if (num == 4) {
        cout << "Muy alto" << endl;
    }
    else if (num == 3) {
        cout << "Alto" << endl;
    }
    else if (num == 2) {
        cout << "Medio" << endl;
    }
    else if (num == 1) {
        cout << "Bajo" << endl;
    }
    else {
        cout << "Valor no válido" << endl;
    }
    return 0;
}
```

Si num == 4 entonces Muy alto

Si num == 3 entonces Alto

Si num == 2 entonces Medio

Si num == 1 entonces Bajo